

Requested Patent: JP4048387A
Title: CIRCUIT DESIGN DEVICE ;
Abstracted Patent: JP4048387 ;
Publication Date: 1992-02-18 ;
Inventor(s): TAKAHASHI SHINICHI; others: 01 ;
Applicant(s): SANYO ELECTRIC CO LTD ;
Application Number: JP19900160273 19900618 ;
Priority Number(s): ;
IPC Classification: G06F15/60 ;
Equivalents: JP2538397B2 ;

ABSTRACT:

PURPOSE: To easily change the content of a symbol library by providing a template part which stores symbol shape data and part of the attribute of the data, and a sheet data part which stores a reference pointer to the template part, connection data, and attribute data changed and added newly.

CONSTITUTION: Circuit design is performed by using symbol information stored in the symbol library 41, and obtained design data is stored in a circuit design data storage part 42. Data read out from the symbol library 41 out of the obtained design data obtained is stored in the template part 42a at every symbol. Attribute data at every symbol stored in the sheet data part 42b is the one only added and changed, therefore, the attribute data at the template part can be used in common to plural symbols. Thereby, even when the amount of design data can be reduced, and also, the content of the symbol library 41 is changed, the content of the design data can be held, and the change of the content of the symbol library 41 can be easily performed.

Japanese Published Unexamined Patent Application (A) No. 04-048387; published February 18, 1992; Application Filing No. 02-160273, filed June 18, 1990; Inventor(s): Shinichi Takahashi et al.; Assignee: Sanyo Electric Corporation; Title of Invention: Circuit Designing Devices

CIRCUIT DESIGNING DEVICES

CLAIM(S)

A circuit designing device comprising: a symbol library for storing the symbol-shaped data for displaying the symbols in the circuit diagram and part of attribute data indicating the attributes of the symbols; a circuit designing data memory section for storing the designing data necessary for specifying the designed circuit containing the symbol-shaped data, the attribute data of each symbol, and the circuit connection data; said circuit designing device being characterized in that said circuit designing data memory section has a template section for storing the symbol-shaped data output from the symbol library and part of the symbol attributes, and a sheet data section for storing a reference pointer for said template section and the circuit connection data as well as the attribute data that have been changed/added.

DETAILED DESCRIPTION OF THE INVENTION

(Field of Industrial Application)

The present invention pertains to a circuit designing device for designing a variety of circuits, particularly to a memory device for the designing data.

(Prior Art)

A circuit designing device, for example, a computer aided device, (CAD), which can design circuits of integrated circuits and of printed circuit boards (PCB) by an interactive process, has conventionally been used. With this device, the operator operates the input device while watching the display on the computer screen, and inputs data to design the circuits.

The circuit designing device has a symbol library for storing the symbol data indicating the symbols used for drafting a circuit diagram and the data regarding the basic attributes of the elements specified by the symbols. When the circuit diagram is drafted, these data are output and used, so the data input can be considerably omitted. Therefore, a circuit can be designed by simple data input.

However, designing a circuit requires the circuit connection data and the attribute data indicating the attributes of symbols, such as a name to

specify each symbol (identifier), a property value (e.g., resistance value), a specific name of every element, and a display position on the circuit diagram. Therefore, the symbol library and the designing data are treated and stored separately.

There are two systems for keeping the designing data. In the first system, when the data stored in the symbol library are retrieved on the circuit diagram, the symbol data stored in the symbol library are all input and stored as the designing data. In the second system, the pointer of the symbol name and specific symbol data are not input.

When the first system is used, the designing data are separated from the data stored in the symbol memory section and the data of the symbol library are not used to draft the circuit diagram. Accordingly, the symbol library and the designing data are completely independent, which makes it easy to manage them. In addition, even when the content in the library is changed, the designing data are not changed, so the designing data used at a time of designing can be preserved without being changed.

In the second system, the data in the symbol library are used as part of the designing data, so the data volume of the designing data can be reduced. Also, changing the content in the symbol library can globally change the symbol content contained in the designing data.

(Problems of the Prior Art to Be Addressed)

According to the aforementioned first system, however, the symbol attribute data in the designing data are separately preserved for every symbol. Therefore, the data volume of the designing data becomes enormously large. Particularly, the designing data volume tends to be increased since various routine checks are done to reassure the perfection of a circuit, and the symbol attributes tend to be increased in order to send the data to the post-process, which also results in a high volume of the designing data. Therefore, there have been demands that the designing data volume should be minimized. In addition, there are cases when the attribute data of one same symbol in the designing data need to be globally changed at one time in the process of circuit designing. In such a case, if the first system is used, every attribute data memorized for every symbol needs to be changed one by one, and the global change is difficult.

With the second system, to save the designing data used at a time of designing without any change, the content in the symbol library at a time of designing needs to be stored as is. But, the content in the symbol library cannot be changed easily, so the data management is difficult, which is a problem.

The present invention, to solve the aforementioned problems, attempts to present a circuit designing device, whereby the symbol attribute data can be globally changed all at one time, while minimizing the designing data volume, and whereby the content in the symbol library can be easily changed.

(Means to Solve the Problems)

The circuit designing device of the present invention comprises: a template section for storing the symbol-shaped data output from the symbol memory section and part of their attributes; a reference pointer for said template section; a sheet data section for storing the circuit connection data and the changed/added attribute data.

(Operation)

The circuit designing device designs the circuit by using the symbol data stored in the symbol library, and the designing data acquired at this time are stored in the data circuit designing data memory section. Of the acquired designing data, the data output from the symbol library are stored in the template for every symbol. The attribute data of every symbol stored in the template section are only those data that are changed/added. Accordingly, the attribute data in the template section can be used in common by multiple symbols.

(Embodiment Example)

The circuit designing device of the present invention is explained below with reference to the drawings.

Fig. 1 shows a block diagram of the structure of the embodiment example of the present invention, wherein the keyboard, input/output device 2, such as a mouse, output devices 3, such as a blotter and a printer, external memory device 4, such as a magnetic disk, and display device 5, such as a CRT, are all connected to CPU 1 which processes the data and controls the input/output.

The external memory 4 comprises: the symbol library 41 for storing the shapes of the symbols (pictures) for displaying the elements, such as various logic circuits, transistors, and resistors, on the circuit diagram; the circuit designing data memory section 42 for storing the designing data (including positions of the elements and the connection relationships as well as the attributes of elements such as the identifiers) created by an interactive method.

The circuit designing data memory section 42, as shown in Fig. 2, consists of template 42a for storing the same data as those in the symbol library and of sheet data section 42a for storing the other data.

In the template section 42a, are written the symbol-shaped data (the pictures of resistor and transistor in this example) input from the symbol library 41 and the basic attributes (In this example, SD = R, which is the symbol identifier meaning a resistor, and MD = NPN, which is the name of the model indicating the type of transistor.).

On the other hand, in the sheet data section 42, are written the position data indicating the positions of the symbols on the circuit diagram, the circuit connection data indicating the state of connections, the attribute data that have been changed/added, and the pointer as a key with which to output the data from the template section 42a. In this example, the data of the circuit consisting of six resistors and of two transistors are stored, and the data indicating the resistance values (150 K, 10 K, 1 K) are added as the attributes. This attribute data are stored in the sheet data section 42b.

When the circuit is designed by this circuit designing device, loading of the program for circuit designing is commanded from the input/output device 2, and the program stored in the external memory device 4 is loaded in the CPU 1 to activate the device. The predetermined circuit designing data are output from many designing files stored in the circuit data memory section 42 of the external memory device 4 and the circuit diagram (the diagram of a printed circuit board) is displayed on the display device 5.

Under this condition, the input/output device 2 is operated to conduct the interactive process, and the circuit is designed by editing it, i.e., by changing, adding, and correcting.

At this time, among the designing data, the attribute data in the template section 42a are not present in the sheet data section 42b. Therefore, if these data need to be displayed, the corresponding data are output from the template section 42a by using the pointer.

In the case when more symbols need to be written in the circuit diagram, the predetermined symbols are designated through the input device 2, and the data are output from the symbol library 41 according to this designation in order to use these data as the designing data. The data output from the symbol library 41 are once input into the template section 42a according to the type of each symbol. Then, in the sheet data section 42b, is written the pointer for specifying the data in the template 42a that corresponds to the position in the circuit diagram designated by the input device 2.

The attributes, such as a resistance value of the symbol and a tolerance value of the resistance, are not included here. Therefore, the operator has to input these attributes of the symbols, in other words, the

operator makes changes to the symbol attribute data by using the input device 2.

Once the attributes of the symbols in the template section 42a are changed, all the attributes of the symbol in the designing data can be changed at one time.

The process of changing the symbol attributes is explained below with reference to the flowchart of Fig. 3.

The process of changing the symbol attributes is input first (S1); the symbols whose attributes are to be changed are designated on the display screen (S2); the command as to whether the change is about the global change (changing the attribute data in the template section 42a) or about the change in the designated symbol (change of the attribute data in the sheet data section 42b) is input (S 3); the result of this input is determined (S 4).

In the case of the global change, the attribute data of the designated symbols in the template section 42a are changed according to the input data (S 5), and these changed data are stored in the template 42a (S 6). Thus, by changing the attribute data in the template, they are reflected as the attribute data of each symbol.

In the case when the attribute data corresponding to the designed symbols are changed, the attribute data corresponding to the designated

symbols in the sheet data section 42b are changed (S 7), and the changed data are stored in the sheet data section 42b (S 8).

In the case when the same attribute data are present in both the sheet data section 42b and template section 42a, the attribute data in the sheet data section 42b have priority over the data in the template section 42a. By this, only the attributes of the specified symbol can be changed.

When the designing data thus created are output to the output device 3, the command for the output is input from the input device 2. Of the designing data, the attribute data that are not changed are stored in the template section 42a. Therefore, by using the pointer in the sheet data section 42b, the symbol-shaped data and their attribute data are output from the template 42a. Then, these data are added to the sheet data and displayed. Also, when the attribute data are output and displayed, these data are likewise output from the template section 42a.

(Advantage)

As explained above, according to the circuit designing device of the present invention, only the changed/added attribute data are stored in the sheet data section, and the data in the template section can be used in common by multiple symbols, which contributes to minimizing the volume of designing data. Also, by changing the data in the template section, the

symbol attributes data can be changed all at one time. In addition, since the data input from the symbol library are stored in the template section, the content of the designing data can be preserved even when the content in the symbol library is updated, so the content in the symbol library can be updated at ease.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a block diagram of the circuit designing device structure as one embodiment example of the present invention.

Fig. 2 illustrates the structures of the sheet data section and of the template section.

Fig. 3 shows a flowchart indicating the steps of changing the attributes.

1. central processing device (CPU)
2. input device
3. output device
4. external memory device
5. display device
41. symbol library
42. circuit designing data memory section

42a. template section

42b. sheet data section

⑫ 公開特許公報(A) 平4-48387

⑤ Int. Cl.⁵

識別記号

庁内整理番号

⑬ 公開 平成4年(1992)2月18日

G 06 F 15/60

3 6 0 A

7922-5L

審査請求 未請求 請求項の数 1 (全5頁)

⑭ 発明の名称 回路設計装置

⑯ 特 願 平2-160273

⑰ 出 願 平2(1990)6月18日

⑱ 発 明 者 高 橋 新 一 大阪府守口市京阪本通2丁目18番地 三洋電機株式会社内
 ⑲ 発 明 者 加 藤 政 美 大阪府守口市京阪本通2丁目18番地 三洋電機株式会社内
 ⑳ 出 願 人 三 洋 電 機 株 式 会 社 大阪府守口市京阪本通2丁目18番地
 ㉑ 代 理 人 弁 理 士 吉 田 研 二 外2名

明 細 書

1. 発明の名称

回路設計装置

2. 特許請求の範囲

回路図中のシンボルを表示するためのシンボル形状データ及びそのシンボルの属性を示す属性データの一部を記憶するシンボルライブラリと、

シンボル形状データ、各シンボルについての属性データ、結線データを含む設計された回路を特定するために必要な設計データを記憶する回路設計データ記憶部を含み、

前記回路設計データ記憶部は、

シンボルライブラリから読み出されたシンボル形状データ及びそのシンボルの属性の一部を記憶するテンプレート部と、

前記テンプレート部への参照ポインタ、結線データ及び変更追加された属性データを記憶するシートデータ部と、

を有することを特徴とする回路設計装置。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、各種回路設計を行う回路設計装置、特に設計データの記憶に関する。

〔従来の技術〕

従来より、IC、PCB(プリント基板)等の回路設計を対話処理によって行える回路設計装置(例えばCAD装置)が利用されている。この回路設計装置では、操作者がディスプレイの表示をみながら入力装置を操作して、データの入力を行い、回路設計を行う。

この回路設計装置では、回路図作成に用いるシンボルを表示するためのシンボル形状データやそのシンボルで特定される素子の基本的な属性等についてのデータを記憶するシンボルライブラリを有している。そして、回路図作成の際には、このデータを読出して利用するため、かなりのデータの入力を省略することができ、簡単な入力により回路設計を行うことができる。

しかし、回路設計のためには、シンボル1つ1つを特定するための名前(識別名)、特性値(例

えば抵抗値等)、素子の具体的な部品名、回路図上における表示位置等そのシンボルの属性を示す属性データや配線データ等の接続情報も必要である。そこで、通常の場合、シンボルライブラリと設計された設計データは別々のものとして存在している。

ここで、この設計データの持ち方として2つの方式がある。第1の方式はシンボルライブラリに記憶されているデータを回路図上に呼び出す場合に、シンボルライブラリに記憶されているシンボル情報をすべて取り込み、これを設計データとして記憶する方式であり、第2の方式はシンボル名前等のポイントだけを取り込み、具体的なシンボル情報は取り込まない方式である。

第1の方式を利用すると、設計データはシンボル記憶部に記憶されているデータとは切り離されたものとなり、回路図を再現するためにシンボルライブラリのデータを必要としない。そこで、シンボルライブラリと設計データの独立性が高まり、これらの管理が容易となる。また、シンボルライ

たい場合が多い。このような場合に、第1の方式では、シンボル毎に記憶されている属性データをすべていちいち変更しなければならず、一括変更が面倒であるという問題点があった。

また、第2の方式では、設計データを設計時点での状態で保存するためには、その時点でのシンボルライブラリの内容が維持されなければならず、シンボルライブラリの内容変更が容易に行えず、その管理が難しいという問題点があった。

本発明は、上記問題点を解決することを課題としてなされたものであり、設計データ量を少なくすると共にシンボル属性の一括変更が簡単に行え、かつシンボルライブラリの内容の変更を容易に行える回路設計装置を提供することを目的とする。

〔課題を解決するための手段〕

本発明に係る回路設計装置は、シンボル記憶部から読み出されたシンボル形状データ及びそのシンボルの属性の一部を記憶するテンプレート部と、前記テンプレート部への参照ポイント、結線データ及び変更追加された属性データを記憶するシー

ブラリの内容を変更しても、設計データは変更されず設計時点の内容をそのまま保持できる。

一方、第2の方式では、設計データの一部としてシンボルライブラリのデータを利用するため、設計データのデータ量を少なくすることができる。更に、シンボルライブラリの内容を変更することで、設計データ中のシンボルの内容を一括変更することができる。

〔発明が解決しようとする課題〕

しかしながら、上記第1の方式によると、設計データのシンボル属性情報をすべてのシンボル毎に別個に保持することとなる。そこで、設計データのデータ量が非常に大きくなってしまふ。特に、回路の完成度をあげるために種々の設計ルールチェックを行ったり、後工程への情報の受け渡しのためにシンボル属性が増加する傾向にあり、設計データ量は増加する傾向にあるため、設計データ量をできるだけ削減したいという要求がある。また、回路設計時においては、設計データ中の同一シンボルについての属性データを一括して変更し

トデータ部と、を有することを特徴とする。

〔作用〕

回路設計装置は、シンボルライブラリに記憶されているシンボル情報を利用して、回路設計を行い、得られた設計データが回路設計データ記憶部に記憶される。ここで、得られた設計データの内の、シンボルライブラリから読み出されたデータはシンボル毎にテンプレート部に記憶される。そして、シートデータ部に記憶されるシンボル毎の属性データは追加変更されたもののみとなっている。このため、テンプレート部における属性データを複数のシンボルに共通のものとして利用できる。

〔実施例〕

以下、本発明に係る回路設計装置について、図面に基づいて説明する。

第1図は、実施例の構成を示すブロック図であり、データの処理、入出力制御等を行うCPU1に、キーボード、マウス等の入力装置2、プロッタ、プリンタ等の出力装置3、磁気ディスク等の外部記憶装置4、CRT等の表示装置5が接続さ

れている。

外部記憶装置4は各種論理回路、トランジスタや抵抗等の素子等の回路図上の表示を行なうためにそれらのシンボルの形状(絵柄)及びそのシンボルの基本的属性を記憶するシンボルライブラリ41及び対話形式によって作成された設計データ(各素子の配置、接続関係の他、各素子についての識別名等の属性を含む)を記憶する回路設計データ記憶部42を含んでいる。

更に、回路設計データ記憶部42は、第2図に示すように、シンボルライブラリと同様の内容のデータを記憶するテンプレート部42aとその他のデータも記憶するシートデータ部42bから構成されている。

そして、テンプレート部42aには、シンボルライブラリ41から取り込まれたシンボル形状データ(この例では抵抗及びトランジスタの絵柄)及び基本的な属性(この例では、抵抗を意味するシンボル識別名であるSD=R、トランジスタを意味する識別名SD=Q及びトランジスタの形式

を示すモデル名MD=NPN)が記述されている。

一方、シートデータ部42bには、回路図上のシンボルの配置位置データ、結線状態を示すデータである結線情報、変更追加分の属性データ、テンプレート部42aにおけるデータ読出しのキーとなるポイントが記述されている。この例では、6つの抵抗と、2つのトランジスタからなる回路についてのデータが記憶され、抵抗値(150k、10k、1k)等を示すデータが属性データとして追加されており、この属性データがシートデータ部42bに記憶されている。

このような回路設計装置により、回路設計を行う場合には、まず入力装置2から回路設計を行うためのプログラムのロードを指令し、外部記憶装置4内に格納されているプログラムをCPU1にロードして装置を起動する。次に、外部記憶装置4の回路データ記憶部42に格納されている各種の設計ファイルの中から所望の回路設計データを読み出して表示装置5上に回路図(プリント基板の回路図)を表示する。

このような状態で、入力装置2を操作し、対話型処理によって回路図の変更、追加、訂正等の編集を行い、回路を設計する。

このとき、設計データの中で、テンプレート部42aにある属性データについては、シートデータ部42bにない。そこで、これらデータの表示等が必要な場合は、シートデータ部42bのポイントに基づき、対応するデータをテンプレート部42aより読み出す。

また、新たにシンボルを回路図上に書き込む場合には、入力装置2により所望のシンボルを指定し、この指定に従ってシンボルライブラリ41内のデータを読み出し、これを設計データとするが、シンボルライブラリ41から読み出されたデータはシンボルの種類ごとにテンプレート部42aに一旦書き込まれる。そして、シートデータ部42bには入力装置2によって指定された回路図上の位置(座標)とテンプレート部42aの対応データを特定するためのポイントが書き込まれる。

ここで、回路図中のシンボルの抵抗値、抵抗値許容差等の属性は入っていない。そこで、これらのシンボルの属性については、入力装置2を利用して操作者がシンボル属性の変更として入力する。

また、テンプレート部42aにおけるシンボルの属性を変更すると、その設計データにおけるそのシンボルの属性を一括変換することができる。

そこで、このシンボル属性の変更について第3図のフローチャートに基づいて説明する。

まず、シンボル属性の変更処理を行うことについての入力を行う(S1)。次に、属性を変更する対象シンボルを画面上で指定する(S2)。そして、一括変更(テンプレート部42aの属性データの変更)か、指定したシンボルの変更(シートデータ部42bにおける属性データの変更)かを入力し(S3)、この入力結果を判定する(S4)。

一括変更の場合には、入力されたデータに従って、テンプレート部42aにおける対象シンボルの対象属性データが変更される(S5)と共に、

これがテンプレート部42aに格納される(S6)。このように、テンプレートの属性を変更することによって、各シンボルの属性データとして反映されることになる。

一方、対象シンボルについてのみの対象属性データの変更である場合には、シートデータ部42bの対象シンボルの対象属性データが変更される(S7)とともに、これがシートデータ部42bに格納される(S8)。

なお、同一属性についてのデータがシートデータ部42b及びテンプレート部42aの両方にある場合には、シートデータ部42bの属性データを優先する。これによって、特定のシンボルのみの属性の変更が有効となる。

そして、このようにして作成された設計データを出力装置3等に出力する場合には、入力装置2により出力指令を入力する。ここで、設計データのうち、変更のない属性データについては、テンプレート部42aに記憶されている。そこで、シートデータ部42bのポインタに基づいてテンプレ

レート部42aのシンボル形状データ及び属性データを読み出し、これをシートデータに加えて出力する。なお、属性データについての表示や出力の場合も同様にテンプレート部42aからデータが読み出される。

[発明の効果]

以上説明したように、本発明に係る回路設計装置によれば、シートデータ部に記憶する属性データは変更追加したもののみであり、テンプレート部のデータを複数のシンボルにおいて共用できるため、設計データの削減を図ることができる。また、テンプレート部のデータを変更することにより、シンボル属性の一括変更も容易に行うことができる。更に、テンプレート部にシンボルライブラリから取り込んだデータを格納しているため、シンボルライブラリの内容更新をしても、設計データの内容を保持することができ、シンボルライブラリの内容変更を容易に行うことができる。

4. 図面の簡単な説明

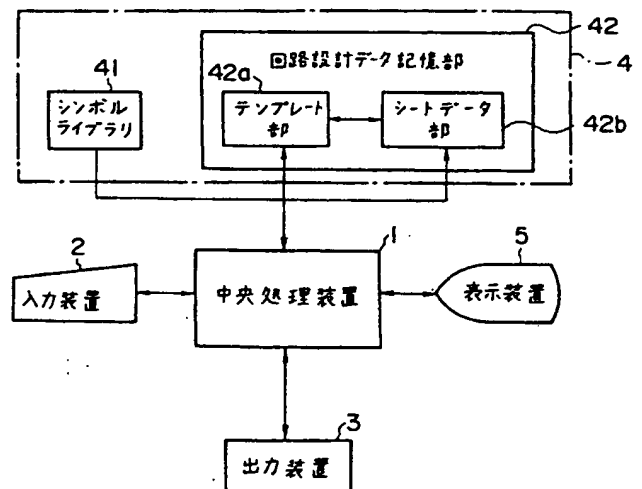
第1図は本発明の実施例に係る回路設計装置の

構成ブロック図、

第2図はシートデータ部とテンプレートの構成を示す説明図、

第3図は属性変更の処理手順を示すフローチャートである。

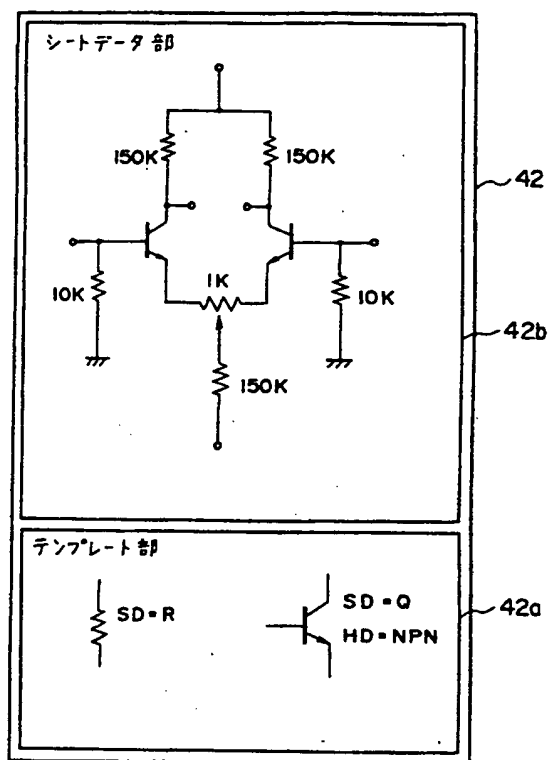
- 1 … 中央処理装置(CPU)
- 2 … 入力装置
- 3 … 出力装置
- 4 … 外部記憶装置
- 5 … 表示装置
- 41 … シンボルライブラリ
- 42 … 回路設計データ記憶部
- 42a … テンプレート部
- 42b … シートデータ部



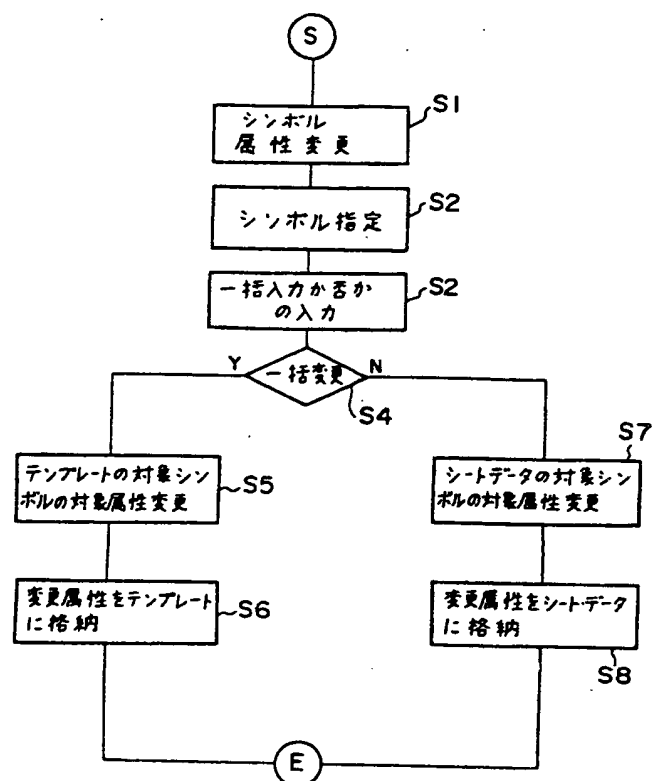
第1図

出願人 三洋電機株式会社
代理人 弁理士 吉田 研二

(外2名) [D-71]



第 2 図



第 3 図